March 30, 2024

## DS503 Big Data Management

## Project 3

Question	Status	Comment
Q1	Fully	Step 1: Creating Dataset
İ	Working	I used the same transaction dataset that I created for Project 1.
		Source Code:
		Project3/Q1_SparkSQL_TransactionDataProcessing/Main.java
		Transactions Dataset (T):
		Project3/Q1_SparkSQL_TransactionDataProcessing/
		Transactions.txt
		Step 2: Spark Workflow
		I completed the workflow in PySpark using Google Collab. I found
		this easiest as I had multiple issues with Scala and my IDE.
		Plan:
		1. Setup Spark
		2. Import transaction data
		3. Add column headers (check schema & dataframe)
		4. Start with T
		5. Create T1
		a. Start with T
		b. Filter
		i. TransTotal >= 200
		6. Create T2
		a. Start with T1
		b. GroupBy TransNumItems
		c. Aggregate
		i. Sum of TransTotal
		ii. Avg of TransTotal
		iii. Min of TransTotal
		iv. Max of TransTotal
		d. OrderBy
		7. Report T2 to client using show
		8. Create T3
		a. Start with T1

	b. GroupBy CustID
	c. Aggregate
	i. Count of CustID
	d. Name it as NumTransT3 for later use
	e. Select
	i. CustID
	ii. NumTransT3
	9. Report T3 to client using show
	10. Create T4
	a. Start with T
	b. Filter
	i. TransTotal >= 600
	11. Create T5
	a. Start with T4
	b. GroupBy CustID
	c. Aggregate
	i. Count of CustID
	d. Name it as NumTransT5 for later use
	e. Select
	i. CustID
	ii. NumTransT5
	12. Report T5 to client using show
	13. Create T6
	a. Create T5 with T3 join
	i. Inner join on CustID
	b. Start with T5 with T3 join
	i. Filter
	1. NumTransT5 * 5 < NumTransT3
	ii. Select
	1. CustID
	14. Report T6 to client using show
	Note: My Transaction Data did not end up having any customers
	in T6, so I tested it with a separate smaller test dataset and
	included the screenshot (T6_verify) of that output to show that it
	does indeed work when there is data that fits that condition.
1	

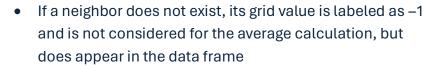
	1	Course Onder
		Source Code:
		Project3/Q1_SparkSQL_TransactionDataProcessing/
		DS503_Project3_Q1.ipynb
		Project3/Q1_SparkSQL_TransactionDataProcessing/
		ds503_project3_q1.py
		Note: Can also be accessed at:
		https://colab.research.google.com/drive/1Xhfhel3j3XklwbhFnQL
		ZrGdkta6QYttO?usp=sharing
		Output with Transaction Dataset:
		Project3/Q1_SparkSQL_TransactionDataProcessing/
		Output_Screenshots/
		Additional Notes:
		<ul> <li>Used SparkSQL &amp; DataFrames to write the workflow</li> </ul>
		Ran into multiple Scala issues because I was unfamiliar
		with the language and my IDE would not cooperate after
		many attempts – my solution was to use PySpark as I am
		familiar with python. Overall, I found that using PySpark
		instead of Scala allowed me to focus more on how to use
		SparkSQL as opposed to having to figure out the details of
		Scala.
Q2	Fully	Step 1: Dataset
	Working	Used the point dataset created in Project 2. I also created a
		smaller dataset with a small number of points in specific areas
		for testing purposes.
		Assumptions:
		All values are integers
		Points Dataset:
		Project3/Q2_SparkRDDs_GridCellsofHighRelativeDensityIndex/
		Points.txt
		Test Point Datasets:
		/Test_Points.txt
		Step 2 & 3: Report the TOP 50 grid cells w.r.t Relative-Density
		Index & Neighbors of the TOP 50 grids
		Plan:
		1. Import points dataset as data frame and set schema
		2. Add column headers
		Assign point's current grid
		o. Assign point a current grid

4. Get count of points in each grid 5. Assign neighbors & check if valid 6. Get count of neighbors 7. For each grid, calculate density 8. Display final outputs a. Top 50 grids with highest relative-density index b. Neighbors of top 50 grids **Source Code:** Test Dataset: Project3/Q2\_SparkRDDs\_GridCellsofHighRelativeDensityI ndex/DS503\_Project3\_Q2\_Test.ipynb Project3/Q2\_SparkRDDs\_GridCellsofHighRelativeDensityl ndex/ds503\_project3\_q2\_test.py Note: Can also be accessed at: https://colab.research.google.com/drive/1Wrx8eRfWII X KAvjBT 4ivy\_2f17AUbV?usp=sharing Points Dataset: Project3/Q2 SparkRDDs GridCellsofHighRelativeDensityI ndex/ DS503\_Project3\_Q2.ipynb Project3/Q2\_SparkRDDs\_GridCellsofHighRelativeDensityI ndex/ds503\_project3\_q2.py Note: Can also be accessed at: https://colab.research.google.com/drive/1rH9bRHKEoD4703EM gYOZAP5uem95zacP?usp=sharing **Output with Test Dataset:** Project3/Q2\_SparkRDDs\_GridCellsofHighRelativeDensityIndex/ Outputs/Test Top50Grid **Output with Points Dataset:** Project3/Q2\_SparkRDDs\_GridCellsofHighRelativeDensityIndex/ Outputs/Points Top50Grid Additional Notes: Grid for reference: 249,501 249,502 250,000 ... 502 1000 501 ...

2

1

500



- Once again used PySpark in google collab as I was unable to get Scala working with my IDE
- Final neighbor output from points dataset took a very long time to run and I did not get to include the final output screenshot because of this, but you should be able to see it through the google collab link